A NEURAL NETWORK FOR DATA ASSOCIATION

Michel Winter and Gerard Favier

Laboratoire I3S CNRS UNSA
2000, route des Lucioles
Les Algorithmes, Bât. Euclide
06410 Sophia-Antipolis
FRANCE
e-mail: winter@i3s.unice.fr

ABSTRACT

This paper presents a new neural solution for solving the data association problem. This problem, also known as the multidimensional assignment problem, arises in data fusion systems like radar and sonar targets tracking, robotic vision... Since it leads to an NP-complete combinatorial optimization, the optimal solution can not be reached in an acceptable calculation time, and the use of approximation methods like the Lagragian relaxation is necessary. In this paper, we propose an alternative approach based on a Hopfield neural model. We show that it converges to an interesting solution that respects the constraints of the association problem. Some simulation results are presented to illustrate the behaviour of the proposed neural solution for an artificial association problem.

1. INTRODUCTION

The multidimensional assignment problem typically arises when several sensors observe the same environment. The data association process have to find which data correspond to the same object. The associations are generally decided in order to minimize a global criterion, using a kind of distance between the data of each sensor. The optimization of this criterion is the aim of this article. This problem also corresponds to an equivalent situation: a sensor observes the environment at several distinct moments. The association problem then consists in finding which data of each scan are generated by the same object. This situation essentially groups sonar and radar targets tracking [7].
If only two sensors are used, some algorithms [1] find the optimal solution in a acceptable calculation time. But when more than two sensors are considered, the problem becomes NP-complete and the optimal solution is not reachable even for a small amount of data. An approximation method is then necessary. Lagrangian relaxation method is generally chosen [2, 3, 4], but we show in this article that a Hopfield neural network is an interesting alternative that delivers an near optimal solution that respect the constraints of the optimization problem.
In the next section, we present the problem formulation and we describe the criterion to be optimized. Section 3 proposes our neural solution to the association problem. Some simulation results are presented in section 4 and concluding remarks are made in section 5.

2. PROBLEM FORMULATION

We define the following notations:

- n the number of sensors (or the number of scans),
- $z^j_{i_j}$ $i_j \in \{0,..,m_j\}$, $j \in \{1,..,n\}$ the i_j-th data of the j-th sensor. We also introduce a *dummy* data z^j_0 that will encode the fact that no data from sensor j is used,
- Z the set of all *real* data, *i.e.* excepted dummy data z^j_0,
- $Z_{i_1 i_2..i_n} = \{z^1_{i_1} z^2_{i_2}, .., z^n_{i_n}\}$ a data association,
- $\Gamma = \{Z_{i_1 i_2..i_n}\}$ a partitioning of the data,

A partitioning Γ is submitted to the following constraints:

(i) a data can not be used twice:

$$\bigcap_{Z_{i_1 i_2..i_n} \in \Gamma} (Z_{i_1 i_2..i_n} \cap Z) = \emptyset \qquad (1)$$

We have to consider the intersection $Z_{i_1 i_2..i_n} \cap Z$ because a dummy association can be used twice.

(ii) each data must be used exactly once:

$$\bigcup_{Z_{i_1 i_2..i_n} \in \Gamma} (Z_{i_1 i_2..i_n} \cap Z) = Z \qquad (2)$$

A cost $c_{i_1 i_2..i_n}$ is also associated to each association $Z_{i_1 i_2..i_n}$. The goal of the multidimensional assignment process is to find the partitioning Γ^* so that:

$$\Gamma^* = arg \min_{\Gamma \in \Omega} \sum_{Z_{i_1 i_2..i_n} \in \Gamma} c_{i_1 i_2..i_n} \qquad (3)$$

where Ω is the set of all possible partitioning Γ, *i.e.* $\Omega = \{\Gamma\}$.
If we define:

$$\rho_{i_1 i_2..i_n} = \begin{cases} 1 & \text{if } Z_{i_1 i_2..i_n} \in \Gamma^* \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

The search of partitioning Γ^* is equivalent to finding the values of the binary variables $\rho_{i_1 i_2..i_n}$ that minimize:

$$\Phi = \sum_{i_1=0}^{m_1} .. \sum_{i_n=0}^{m_n} \rho_{i_1 i_2..i_n} c_{i_1 i_2..i_n} \quad (5)$$

under the constraints:

$$\begin{cases} \sum_{i_2=0}^{m_2} .. \sum_{i_n=0}^{m_n} \rho_{i_1..i_n} = 1 & \forall i_1 \in \{1..m_1\} \\ \sum_{i_1=0}^{m_1} \sum_{i_3=0}^{m_3} .. \sum_{i_n=0}^{m_n} \rho_{i_1..i_n} = 1 & \forall i_2 \in \{1..m_2\} \\ \vdots \\ \sum_{i_1=0}^{m_1} .. \sum_{i_{n-1}=0}^{m_{n-1}} \rho_{i_1..i_n} = 1 & \forall i_n \in \{1..m_n\} \end{cases} \quad (6)$$

another expression of the constraints can be: $\forall \rho_{i_1..i_n}$

$$\begin{aligned} &\rho_{i_1..i_n} = 1 \Longleftrightarrow \\ &(1-\delta_{i_1 0}) \sum_{i'_2=0}^{m_2} .. \sum_{i'_n=0}^{m_n} \left(1 - \prod_{j=2}^{n} \delta_{i_j i'_j}\right) \rho_{i_1 i'_2..i'_n} \\ &+ .. \\ &+ (1-\delta_{i_n 0}) \sum_{i'_1=0}^{m_1} .. \sum_{i'_{n-1}=0}^{m_{n-1}} \left(1 - \prod_{j=1}^{n-1} \delta_{i_j i'_j}\right) \rho_{i'_1..i'_{n-1} i_n} = 0 \end{aligned} \quad (7)$$

where δ_{ij} is the Kronecker symbol:

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Equation 7 can be rewritten as :

$$\begin{aligned} &\rho_{i_1..i_n} = 1 \Longleftrightarrow \quad (9) \\ &\sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} \sum_{j=1}^{n} \delta_{i_j i'_j} \left(1 - \delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l \neq j}}^{n} \delta_{i_l i'_l}\right) \rho_{i'_1 ... i'_n} = 0 \end{aligned}$$

3. NEURAL OPTIMIZATION

We propose to develop a Hopfield neural network that approximates a near optimal solution of the minimization of criterion 5 under constraints 9. The network minimizes the following energy :

$$\begin{aligned} E_H = &\sum_{i_1=0}^{m_1} .. \sum_{i_n=0}^{m_n} c_{i_1...i_n} f(x_{i_1.i_n}) \quad (10) \\ &+ \frac{1}{2} \sum_{i_1=0}^{m_1} .. \sum_{i_n=0}^{m_n} \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} W_{i'_1..i'_n}^{i_1..i_n} f(x_{i_1..i_n}) f(x_{i'_1..i'_n}) \\ &+ \sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \int_0^{x_{i_1..i_n}} \varepsilon f'(\varepsilon) d\varepsilon \end{aligned}$$

where f is the sigmoid function:

$$f(x) = \left(\frac{1}{1 + e^{-4px}} \right) \quad (11)$$

where p is the slope of the function. The weights $W_{i'_1..i'_n}^{i_1..i_n}$ are given by:

$$W_{i'_1..i'_n}^{i_1..i_n} = C \sum_{j=1}^{n} \left[\delta_{i_j i'_j} \left(1 - \delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l \neq j}}^{n} \delta_{i_l i'_l}\right) \right] \quad (12)$$

where C is a positive parameter. Each possible association $\rho_{i_1...i_n}$ is represented by a neuron denoted $x_{i_1...i_n}$. At convergence, the values of the variables $\rho_{i_1...i_n}$ is given by the outputs of the neurons, *i.e.* $\rho_{i_1...i_n} = f(x_{i_1..i_n})$.

The first term of energy E_H clearly corresponds to the minimization of the criterion 5. The second term is associated to the constraints 9. The last term is necessary to ensure that the network converges properly, but its influence is generally neglected by considering that the sigmoid function is sharp enough.
In the discrete time case, the neuron update rule that ensures the minimization of E_H is given by [5]:

$$x_{i_1..i_n}(k) = -c_{i_1..i_n} - \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} W_{i'_1..i'_n}^{i_1..i_n} f(x_{i'_1..i'_n}(k-1)) \quad (13)$$

We are now going to proove that the solution obtained at the convergence verifies the constraint 7, under the condition:

$$-1 < c_{i_1...i_n} < 0 \quad (14)$$

It is shown in appendice A that it is always possible to transform the costs $c_{i_1...i_n}$ so that they verify 14, without changing the optimization process; thus, we will consider in the follwing that 14 is respected.
Theorem 1: *neurons states $x_{i_1...i_n}$ converge.*
Proof: we know [6] that neurons outputs converge, *i.e.*:

$$\lim_{k \to \infty} f(x_{i_1...i_n}(k)) = y^0_{i_1...i_n} \qquad , \forall i_1, .., i_n \quad (15)$$

where $y^0_{i_1...i_n}$ represents the ouput of the $i_1 ... i_n$-th neuron, at convergence. Using 13, we can write :

$$\begin{aligned} \lim_{k \to \infty} \quad & x_{i_1...i_n}(k) \\ &= \lim_{k \to \infty} \left[-c_{i_1..i_n} - \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} W_{i'_1..i'_n}^{i_1..i_n} f(x_{i'_1..i'_n}(k)) \right] \\ &= \lim_{k \to \infty} \left[-c_{i_1..i_n} - \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} W_{i'_1..i'_n}^{i_1..i_n} y^0_{i'_1..i'_n} \right] \\ &= x^0_{i_1..i_n} \qquad , \forall i_1, .., i_n \end{aligned} \quad (16)$$

where $x^0_{i_1\ldots i_n}$ is the neuron state at convergence.

Theorem 2: *neurons outputs converge to values in* $\{0, 0.5, 1\}$ *when the slope* p *of the sigmoid function tends toward infinity.*
Proof: we have:

$$y^0_{i_1\ldots i_n} = f(x^0_{i_1\ldots i_n}) = \frac{1}{1+e^{-4px^0_{i_1\ldots i_n}(k)}} \quad (17)$$

and:

$$\lim_{p\to\infty} f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 0.5 & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases} \quad (18)$$

using 17 and 18, proof is evident.

Theorem 3: *neurons outputs converge to values in* $\{0, 1\}$ *if* $C > 2$.
Proof: we have to show that the case $x^0_{i_1\ldots i_n} = 0$ is impossible. At convergence, 13 can be rewritten as follows :

$$x^0_{i_1\ldots i_n} = -c_{i_1..i_n} - \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} W^{i_1..i_n}_{i'_1..i'_n} y^0_{i'_1..i'_n} \quad (19)$$

We prooved that $y^0_{i'_1\ldots i'_n} \in \{0, 0.5, 1\}$. Moreover, the weights described in 12 equals 0 or C, thus 19 leads to:

$$x^0_{i_1\ldots i_n} = -c_{i_1\ldots i_n} - \frac{\alpha C}{2} = 0 \qquad \text{with } \alpha \in I\!N^+ \quad (20)$$

So, if $C > 2$ it is easy to see that $x^0_{i_1\ldots i_n} = 0$ is impossible.

Theorem 4: *solution given by the neural network at convergence verifies the constraint* 7.
Proof: we denote A the proposition "$y^0_{i_1\ldots i_n} = 1$", and B the proposition:

$$\sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} \sum_{j=1}^{n} \delta_{i_j i'_j} \left(1-\delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l\neq j}}^{n} \delta_{i_l i'_l}\right) y^0_{i'_1\ldots i'_n} = 0 \quad (21)$$

Constraint 7 is verified if we have:

$$A \Longleftrightarrow B \quad (22)$$

or:

$$B \Rightarrow A \quad (23)$$

$$\text{and} \quad \overline{B} \Rightarrow \overline{A} \quad (24)$$

where $\overline{A}$ represents the proposition $y^0_{i_1\ldots i_n} = 0$, and $\overline{B}$ is:

$$\sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} \sum_{j=1}^{n} \delta_{i_j i'_j} \left(1-\delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l\neq j}}^{n} \delta_{i_l i'_l}\right) y^0_{i'_1..i'_n} > 0 \quad (25)$$

First, we show that $B \Rightarrow A$:

$$B \Longleftrightarrow \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} \sum_{j=1}^{n} \delta_{i_j i'_j} \left(1-\delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l\neq j}}^{n} \delta_{i_l i'_l}\right) y^0_{i'_1..i'_n} = 0$$

Multiplying by a constant $C \neq 0$ leads to:

$$B \Rightarrow \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} \sum_{j=1}^{n} C\delta_{i_j i'_j} \left(1-\delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l\neq j}}^{n} \delta_{i_l i'_l}\right) y^0_{i'_1..i'_n} = 0$$

$$\Rightarrow \sum_{i'_1=0}^{m_1} \ldots \sum_{i'_n=0}^{m_n} W^{i_1\ldots i_n}_{i'_1\ldots i'_n} y^0_{i'_1\ldots i'_{n-1} i_n} = 0 \quad (26)$$

using 19 and 14, this leads to:

$$B \Rightarrow x^0_{i_1\ldots i_n} = -c_{i_1\ldots i_n} > 0 \quad (27)$$

and:

$$B \Rightarrow y^0_{i_1\ldots i_n} = 1 \quad (28)$$

showing that $B \Rightarrow A$. We must now verify that $\overline{B} \Rightarrow \overline{A}$:

$$\overline{B} \Longleftrightarrow \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} \sum_{j=1}^{n} \delta_{i_j i'_j} \left(1-\delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l\neq j}}^{n} \delta_{i_l i'_l}\right) y^0_{i'_1..i'_n} > 0$$

We prooved that $y^0_{i'_1\ldots i'_n} \in \{0, 1\}$; so, multiplying by a constant $C \neq 0$ leads to:

$$\overline{B} \Rightarrow \sum_{i'_1=0}^{m_1} .. \sum_{i'_n=0}^{m_n} \sum_{j=1}^{n} C\delta_{i_j i'_j} \left(1-\delta_{i_j 0}\right) \left(1 - \prod_{\substack{l=1 \\ l\neq j}}^{n} \delta_{i_l i'_l}\right) y^0_{i'_1..i'_n} => C$$

adding $c_{i'_1\ldots i'_n}$ and multiplying by -1 show that:

$$\overline{B} \Rightarrow -c_{i_1\ldots i_n} - \sum_{i'_1=0}^{m_1} \ldots \sum_{i'_n=0}^{m_n} W^{i_1\ldots i_n}_{i'_1\ldots i'_n} y^0_{i'_1\ldots i'_n} <= -c_{i_1\ldots i_n} - C$$

if $C > 2$ we can write:

$$\overline{B} \Rightarrow x^0_{i_1\ldots i_n} < 0 \quad (29)$$

$$\Rightarrow y^0_{i_1\ldots i_n} = 0 \quad (30)$$

which shows that $\overline{B} \Rightarrow \overline{A}$.

4. SIMULATIONS RESULTS

We made some simulations in order to examine the behaviour of the proposed neural method. We consider four sensors delivering points in the 2 dimensional plan. The goal of the data association is to form figures with 2, 3 or 4 points, each figure containing at most one data of each sensor. The cost of each association is the perimeter of the figure divided by its number of segments (*i.e.* 1 segment for a 2 points association, 3 segments for a 3 points association and 4 segments for a 4 points association). We have generated 50 different association problems and we have computed the optimal solution for each of them. 17.4% of the solutions given by our network are optimal ones, and the mean error between the optimal solution and the obtained solution equals 2.7%. 100% of the obtained solutions verify the constraints.

5. CONCLUSION

We proposed a neural approach to solve the data association problem that arises in data fusion systems or in radar [7] and sonar [2] tracking. We prooved that the obtained solution always verifies the constraints of the problem, and some simulations showed that our approach always converges to a near optimal solution.

6. REFERENCES

[1] F. Burgeois and J. C. Lassalle, "An extension of the munkres algorithm for the assignment problem to rectangular matrices," in *Communications of the ACM*, vol. 14, pp. 802–806, December 1971.

[2] H. Gauvrit, C. Jauffret, and J.-P. L. Cadre, "Application de l'optimisation combinatoire a l'extraction-poursuite de sources multiples," in *Proceedings of the GRETSI'97 Conference*, vol. 2, pp. 1129–1132, September, Grenoble FRANCE 1997.

[3] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Transactions on Automatic Control*, vol. AC-37, pp. 198–213, February 1992.

[4] A. B. Poore and N. Rijavec, "Multi-target tracking and multidimensional assignment problems," in *Proceedings of the SPIE Signal and Data Processing of Small Targets*, vol. 1481, April, Orlando 1991.

[5] J. J. Hopfield, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, pp. 141–152, 1985.

[6] P. Koiran, "Dynamics of discrete time, continous state hopfield networks," *Neural Computation*, vol. 6, pp. 459–468, 1994.

[7] M. Winter and G. Favier, "A neural solution for multitarget tracking based on a maximum likelihood approach," in *Proceedings of ICASSP'98*, Seattle, March 1998.

A. COSTS NORMALIZATION

We show in this section that it is possible to change the costs $c_{i_1..i_n}$ into the costs $\mathcal{C}_{i_1..i_n} = c_{i_1..i_n} + \sum_{j=1}^{n} a_{i_j}^j$, with $a_0^j = 0 \ \ \forall j \in \{1,..,n\}$, without changing the optimization process. We define:

$$\mathcal{J} = \sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \mathcal{C}_{i_1..i_n} \rho_{i_1..i_n} \tag{31}$$

Expression of criterion $\mathcal{J}$ can be rewritten as follows:

$$\begin{aligned} \mathcal{J} &= \sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \mathcal{C}_{i_1..i_n} \rho_{i_1..i_n} \\ &= \sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \left(c_{i_1..i_n} + \sum_{j=1}^{n} a_{i_j}^j \right) \rho_{i_1..i_n} \\ &= \Phi + \sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \left(\sum_{j=1}^{n} a_{i_j}^j \right) \rho_{i_1..i_n} \end{aligned} \tag{32}$$

We must show that the second term of 32 is constant. We have:

$$\sum_{i_2=0}^{m_1} ... \sum_{i_n=0}^{m_n} \rho_{i_1..i_n} = 1 \qquad \forall\, i_1 \in \{1,..,m_1\} \tag{33}$$

and so:

$$\sum_{i_2=0}^{m_1} ... \sum_{i_n=0}^{m_n} \rho_{i_1..i_n} a_{i_1}^1 = a_{i_1}^1 \qquad \forall\, i_1 \in \{1,..,m_1\} \tag{34}$$

We have $a_0^j = 0$, thus we can write:

$$\sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \rho_{i_1..i_n} a_{i_1}^1 = \sum_{i_1=0}^{m_1} a_{i_1}^1 \tag{35}$$

taking into account each sensor, we have:

$$\left\{ \begin{array}{l} \sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \rho_{i_1..i_n} a_{i_1}^1 = \sum_{i_1=0}^{m_1} a_{i_1}^1 \\ ... \\ \sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \rho_{i_1..i_n} a_{i_n}^n = \sum_{i_n=0}^{m_n} a_{i_n}^n \end{array} \right. \tag{36}$$

and:

$$\sum_{i_1=0}^{m_1} ... \sum_{i_n=0}^{m_n} \left(\sum_{j=1}^{n} a_{i_j}^j \right) \rho_{i_1..i_n} = \sum_{l=1}^{n} \sum_{i_l=0}^{m_l} a_{i_l}^l \tag{37}$$

which is a constant. This result shows that it is possible to add constants terms to each costs without changing the minimization process. By using this result and by dividing the costs by a normalization factor, it is always possible to consider costs that verify 14.